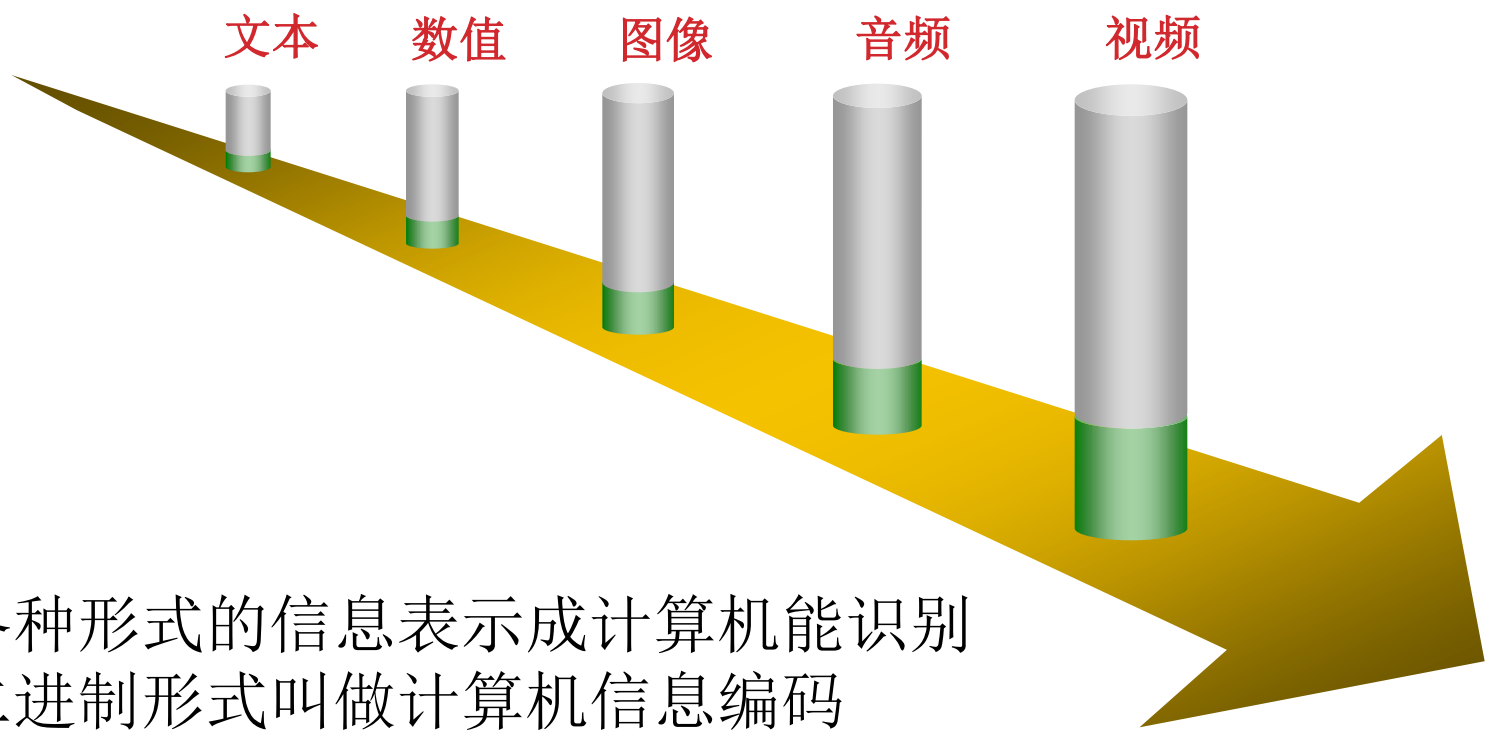
第十二课 数据的表示和运算

对外经济贸易大学信息学院

本章目标

- 理解不同类型数据是如何在计算机内进行表示和存储的
- 理解不同进制数据是如何进行相互转换的
- 理解整数和浮点数表示
- 理解二进制逻辑运算

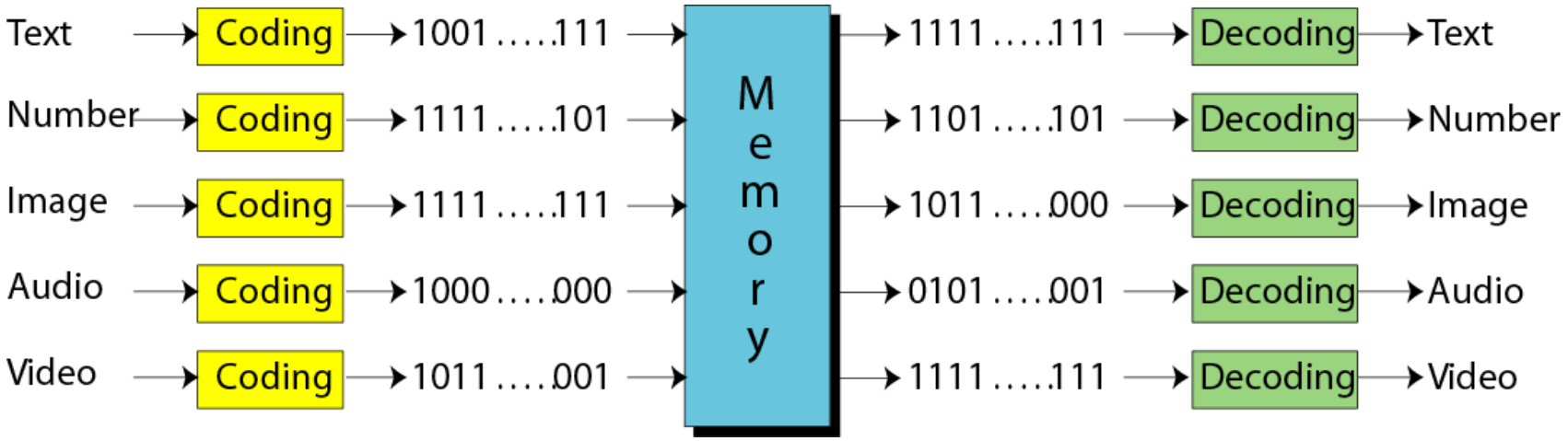
数据的类型



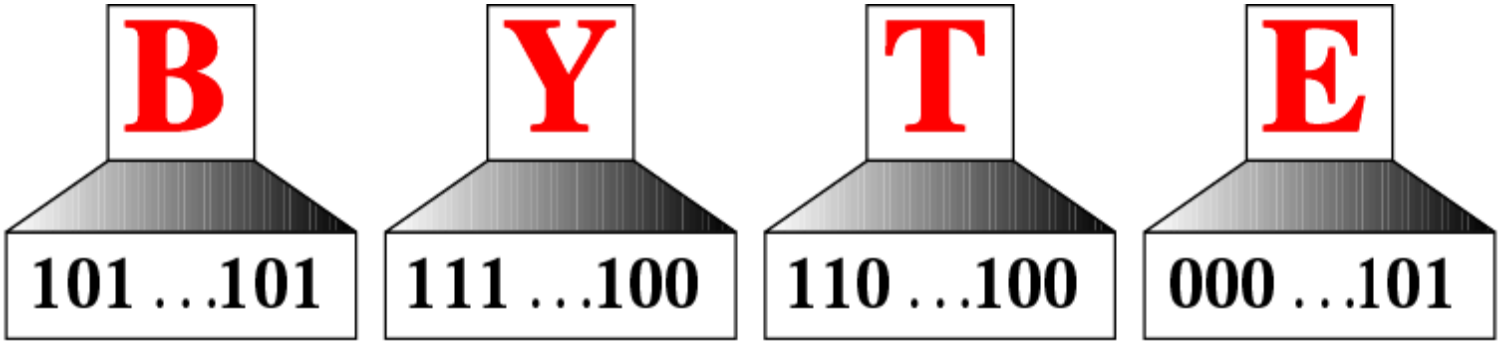
将各种形式的信息表示成计算机能识别的二进制形式叫做计算机信息编码

数据的存储

- 位模式：位的序列



数据的表示



文本的表示

字符编码

- ASCII（American Standard Code for Information Interchange）
  - 美国(国家)信息交换标准代码
  - 标准ASCII码为7位，扩充为8位－－1个字节
  - 最多可以编码256个字符
  - 包括 字母、数字、标点符号、控制字符及其他符号分配数值

常见ASCII码

常用字符的ASCII值表

值	符号	值	符号	值	符号
0	空字符	44	,	91	[
32	空格	45	-	92	\
33	!	46	.	93	]
34	"	47	/	94	^
35	#	48～57	0～9	95	_
36	$	58	:	96	`
37	%	59	;	97～122	a～z
38	&	60	<	123	{
39	'	61	=	124	|
40	(	62	>	125	}
41	)	63	?	126	~
42	*	64	@	127	DEL(delete键)
43	+	65～90	A～Z		

汉字编码

汉字编码

- GB2312
  - 汉字字符集国家标准编码
  - 简体中文字符集，由6763个常用汉字和682个全角的非汉字字符组成
  - 2个字节(16位)表示一个汉字
- GBK
  - 汉字内码扩展规范
  - GB2312的扩展，加入对繁体字的支持
  - 2个字节表示一个汉字
- GB18030
  - 解决了中文、日文、朝鲜语等的编码，兼容GBK
  - 变字节表示(1 ASCII，2，4字节表示字)

通用编码

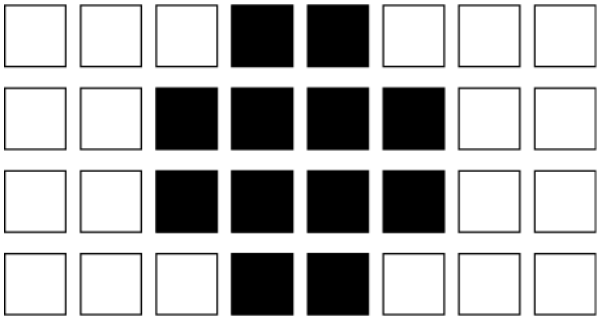
- Unicode
  - 世界650种语言进行统一编码
  - 多个编码方式，分别是UTF-8，UTF-16和UTF-32

汉字的编码

- 交换码（国际码）
  - 汉字信息交换的标准编码
- 外码（输入码）
  - 使用英文键盘输入汉字时的编码
- 机内码
  - 汉字在计算机中存储使用的编码
  - 采用变形国标码
  - 对应GB2312采用2个字节
- 字形码
  - 显示和打印汉字的编码
  - 通常用16×16点阵来显示汉字

图像的表示

- **位映射图像**
  - 以点阵形式存取文件，读取时候按点排列顺序读取数据
  - 像素：阵列中的点
- 例子：黑白图像



0 0 0 1 1 0 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 0 1 1 0 0 0

Image

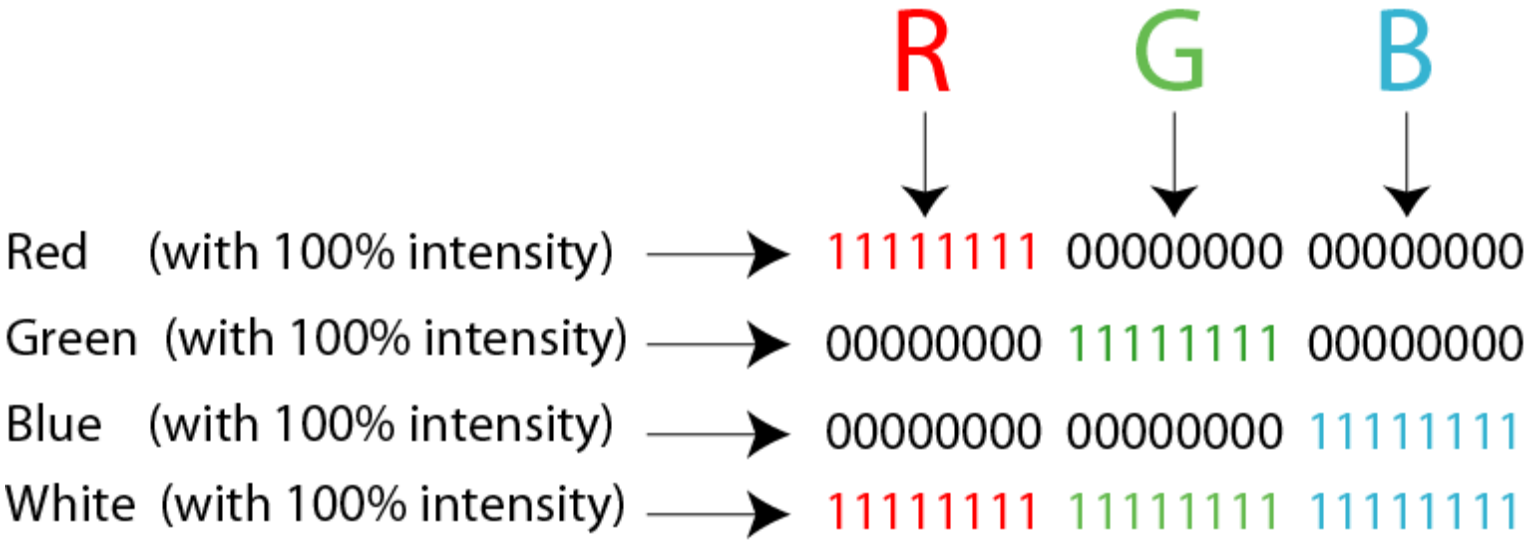
Matrix Representation

00011000 00111100 00111100 00011000

Linear Representation

图像的表示

- 例子：彩色图像
  - RGB：红、绿、蓝三原色；显示器；
  - CMYK：青、品红、黄、黑；彩色印刷；
  - 色彩深度（色彩位数）
    - 位图中使用二进制表示每个点颜色的位数（24位，真彩色）



图像的表示

- 矢量图
  - 使用直线和曲线来描述图形，基本元素是一些点、线、矩形、多边形、圆和弧线等
  - 通过数学公式计算获得
  - 例如：
    - 一幅花的矢量图形实际上是由线段形成外框轮廓，由外框的颜色以及外框所封闭的颜色决定花显示出的颜色。
  - 优点：
    - 可通过公式计算获得，所以文件体积一般较小
    - 无论放大、缩小或旋转等不会失真
  - 缺点
    - 难以表现色彩层次丰富的逼真图像效果

图像的表示

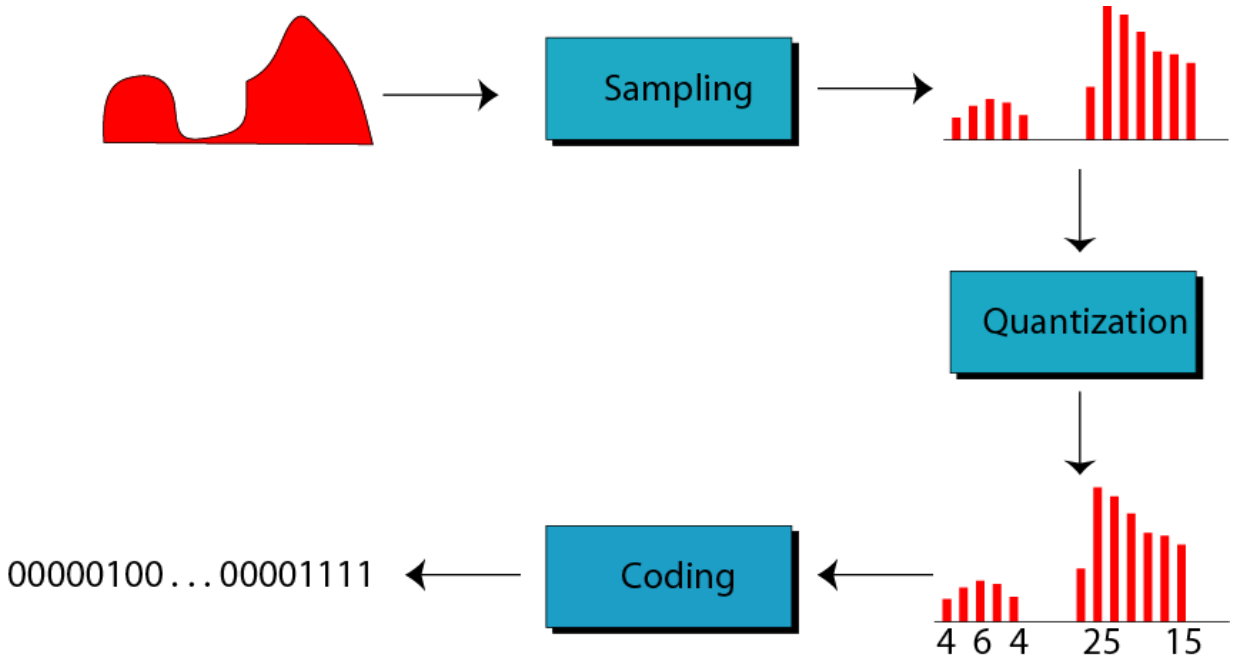
- 例子：

常见的图像文件格式

- BMP
  - Windows系统下的标准位图格式
  - 未经过压缩，一般图像文件会比较大
- JPEG
  - 特殊的有损压缩算法，将不易被人眼察觉的图像颜色删除，从而达到较大的压缩比
- PNG
  - Portable Network Graphics可移植的网络图形
  - 支持图像透明
- GIF
  - Graphics Interchange Format图像互换格式
  - 经过压缩最多支持256种色彩的图像
  - 文件中可以存多幅彩色图像，可构成一种最简单的动画
- AI、EPS
  - 矢量图格式

音频的表示

- 音频
  - 模拟数据采样称离散的数字信号
  - 量化后用二进制存贮



视频的表示

- 视频
  - 图像（帧）在时间上的表示
  - 本质是一系列连续播放而形成的运动图像

数据制式的转换

数码、基数、权值

十进制

- 十进制（Decimal）
  - 数码
    - 0、1、2、3、4、5、6、7、8、9
  - 表示
    - 5029、5029D、$(5029)_{10}$
    - 649.37、649.37D、$(649.37)_{10}$
  - 加法“逢10进1”；减法“借1当10”；
  - 进位计数制
    - 基数：10
    - $5029=5*10^3+0*10^2+2*10^1+9*10^0$
    - $649.37=6*10^2+4*10^1+9*10^0+3*10^{-1}+7*10^{-2}$

二进制

- 二进制（Binary）
  - 数码
    - 0、1
  - 表示
    - 10101B、$(10101)_2$
    - 1101.01B、$(1101.01)_2$
  - 加法“逢2进1”；减法“借1当2”；
    - 110B+10B=1010B
  - 进位计数制
    - 基数：2
    - $10101B=1*2^4+0*2^3+1*2^2+0*2^1+1*2^0$
    - $1101.01B=1*2^3+1*2^2+0*2^1+1*2^0+0*2^{-1}+1*2^{-2}$

八进制

- 八进制（Octal）
  - 数码
    - 0、1、2、3、4、5、6、7
  - 表示
    - 237O、$(237)_8$
    - 42.17O、$(42.17)_8$
  - 加法“逢8进1”；减法“借1当8”；
    - 641O - 74O = 545O
  - 进位计数制
    - 基数：8
    - $237=2*8^2+3*8^1+7*8^0$
    - $42.17=4*8^1+2*8^0+1*8^{-1}+7*8^{-2}$

十六进制

- 十六进制（Hexadecimal）
  - 数码
    - 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F
  - 表示
    - 8F1EH、$(8F1E)_{16}$
    - 1A.3CH、$(1A.3C)_{16}$
  - 加法“逢16进1”；减法“借1当16”；
    - 25H + A1H = C6H
  - 进位计数制
    - 基数：16
    - $8F1EH=8*16^3+15*16^2+1*16^1+14*16^0$
    - $1A.3CH=1*16^1+10*16^0+3*16^{-1}+12*16^{-2}$

与十进制的转换

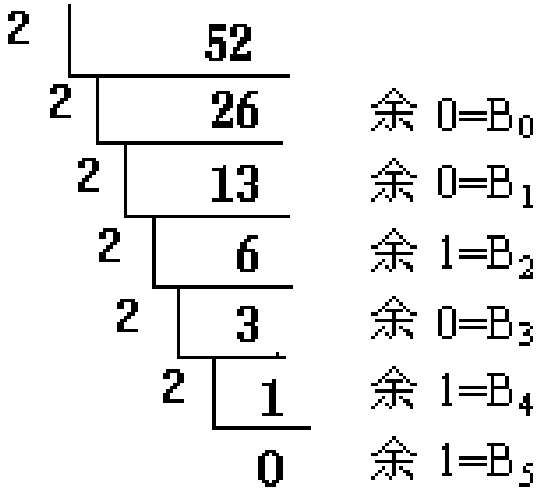
- 转换为十进制
  - 采用进位计数制运算
  - 注意权值

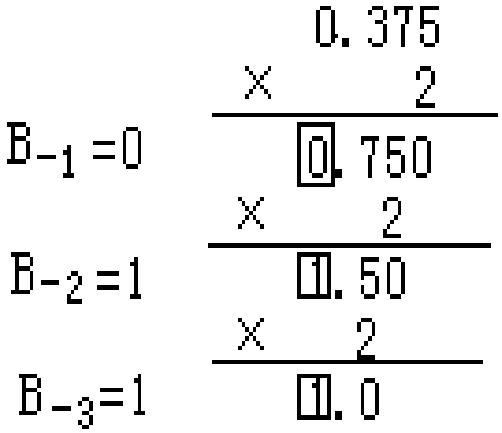
例子：

- $(11000.101)_2$
  - $=1\times2^4+1\times2^3+0\times2^2+0\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=(24.625)_{10}$
- $(103)_8$
  - $=1\times8^2+0\times8^1+3\times8^0=（67）_{10}$
- $(B5.2)_{16}$
  - $=11\times16^1+5\times16^0+2\times16^{-1}=(181.125)_{10}$

十进制到二进制的转换

- 十进制转换为二进制
  - 整数部分：除2取余，从低位开始
  - 小数部分：乘2取整，从高位开始
- 例子：52.375=110100.011B





十进制到八进制的转换

- 十进制转换为八进制
  - 整数部分：除8取余，从低位开始
  - 小数部分：乘8取整，从高位开始
- 例：150.12=226.07534O

0.12*8=0.96,取0

0.96*8=7.68,取7

0.68*8=5.44,取5

0.44*8=3.52,取3

0.52*8=4.16,取4

二进制与八进制

- 八进制→二进制
  - 八进制的每1位用3位二进制表示
  - 31.2O = 011 001 . 010B
- 二进制→八进制
  - 从小数点起，向左、右每3位二进制用一位八进制表示（不足3位用0补齐）
    - 1101100.1B = 154.4O

二进制与十六进制

- 十六进制→二进制
  - 十六进制的每1位用4位二进制表示
  - 1B3.2H = 0001 1011 0011 . 0010B
- 二进制→十六进制
  - 从小数点起，向左、右每4位二进制用一位十六进制表示（不足4位用0补齐）
    - 1101100.1B = 6C.8H

常用的进制转换

十进制数	二进制数	八进制数	十六进制数
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

数的存储

整数（无符号、有符号）、浮点数

整数的存储

- 无符号整数
  - 没有符号的整数，直接表示成二进制
  - 范围
    - $0 \sim 2^N-1$；例如，8位二进制 [0, 255]

Decimal	*8-bit allocation*	*16-bit allocation*
7	00000111	0000000000000111
234	11101010	0000000011101010
258	overflow	0000000100000010
24,760	overflow	0110000010111000
1,245,678	overflow	overflow

整数的存储

- 有符号整数
  - 原码
    - 符号加绝对值格式
  - 反码
    - 二进制反码格式
  - 补码
    - 二进制补码格式

整数的存储

- 原码
  - 符号位加上真值的绝对值
  - 第一位表示符号（0表示正数，1表示负数），其余位表示数值
  - 范围
    - $-2^{N-1}+1 \sim 2^{N-1}-1$，例如8位二进制 [-127, 127]

Decimal	*8-bit allocation*	*16-bit allocation*
+7	00000111	0000000000000111
-124	11111100	1000000001111100
+258	overflow	0000000100000010
-24,760	overflow	1110000010111000

整数的存储

- 反码
  - 正数的反码和原码相同；
  - 负数的反码是原码的符号位不变其余位取反；
  - 0有两种表示方法：全0或者全1
  - 范围
    - $-2^{N-1}+1 \sim 2^{N-1}-1$，例如8位二进制 [-127, 127]

Decimal	***8-bit allocation***	***16-bit allocation***
+7	00000111	0000000000000111
-7	11111000	1111111111111000
+124	01111100	0000000001111100
-124	10000011	1111111110000011
+24,760	overflow	0110000010111000

整数的存储

- 补码
  - 正数的补码和原码相同；
  - 负数的补码是 反码＋1；
  - 范围
    - $-2^{N-1} \sim 2^{N-1}-1$，例如8位二进制 [-128, 127]

Decimal	*8-bit allocation*	*16-bit allocation*
+7	00000111	0000000000000111
-7	11111001	1111111111111001
+124	01111100	0000000001111100
-124	10000100	1111111110000100
+24,760	overflow	0110000010111000

浮点数的表示

- 浮点数的表示
  - 两部分：整数部分＋小数部分
  - 规范化：移动浮点数的小数点使小数点的左边只有一个“1”；

Original Number	***Move***	***Normalized***
+1010001.1101	← 6	$+2^{6}$ x 1.01000111001
-111.000011	← 2	-2^{2} x 1.11000011
+0.00000111001	6 →	$+2^{-6}$ x 1.11001
-0.001110011	3 →	-2^{-3} x 1.110011

浮点数的表示

- 规范化后：
  - 符号
  - 阶码
  - 尾数
- 例子：

阶码

符号

$+2^{6}$ x 1.01000111001

尾数

- 浮点数标准
  - 单精度：32位（4个字节）
  - 双精度：64位（8个字节）

数的运算

算数运算、逻辑运算

算数运算

- 加
- 减
- 乘
  - 采用连加的方式
- 除
  - 采用连减的方式
- ！ 数据以**补码**形式进行存储！

算数运算

- 加法运算
  - 两个补码形式二进制数直接相加
  - （问题：两个正整数相加却得到负数？溢出）
- 减法运算
  - 将减数取反，转换成加法运算

逻辑运算

- 位表示逻辑
  - 1表示“真”
  - 0表示“假”
- 逻辑运算
  - 非
  - 与
  - 或
  - 亦或

真值表

NOT

x	NOT x
0	1
1	0

AND

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

XOR

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	0

练习

1. 将121.375分别表示为二进制、八进制、十六进制数；
2. 将无符号整数11010101表示为十进制数；
3. 将有符号补码11010101表示为十进制数；
4. 将-67表示为8位二进制补码形式；
5. 采用二进制方式计算98+(-67)；
6. 逻辑运算 10110101 OR 10010111